OAK RIDGE NATIONAL LABORATORY
operated by
UNION CARBIDE CORPORATION
NUCLEAR DIVISION
for the
U.S. ATOMIC ENERGY COMMISSION

ORNL-TM-1795

COPY NO. - 69

DATE - March 22, 1967

"BIG HOOP" AND "LITTLE HOOP" -- TWO "PETER T. HOOPER" ROUTINES FOR CHECKING THE DATA BREAK ON A PDP-8

B. W. Rust† and W. R. Burrus

ABSTRACT

Two routines have been written to check the data break for the PDP-8. LITTLE HOOP flickers for each event a dot on the oscilloscope with x and y coordinates equal to any two of the four signals for that event. BIG HOOP does the same buit, unlike LITTLE HOOP which requires the user to specify which two signals he wants plotted at the start of the program, BIG HOOP also contains a command program COMMD, which allows the user to change the signal being plotted along each axis by typing in changes from the keyboard. COMMD also can serve as a prototype for more complex and powerful command routines which could be used to give the experimenter more flexibility in controlling what the computer does during the experiment.

This Work Supported by
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Under Order R-104(1)

†Computer Technology Center, Oak Ridge Gaseous Diffusion Plant.

NOTICE This document contains information of a preliminary nature and was prepared primarily for internal use at the Oak Ridge National Laboratory. It is subject to revision or correction and therefore does not represent a final report.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

9/2/1

BRIEF DESCRIPTION OF THE DATA BREAK CONTROL

Data from the pulse-height analyzers enter the interface through the data break control which multiplexes 48 bits into four successive computer words. Words are stacked in a buffer region of core whose origin and length are controlled by hardware switches. The buffer region can be either 256_{10} (400_8) or 512_{10} (1000_8) words long. If the length switches are set for a 256-word buffer then the origin switches can be set for any of the locations 0000, 0400, 1000, ..., 7400. For a 512-word buffer the possible origins are 0000, 1000, 2000, ..., 7000. The programs described in this report are written for a 256-word buffer beginning in core location 1000. Switches are also available for controlling the number of words per event and these can be set for any of the numbers 1, 2, 3, or 4. These programs are designed for 4 words per event. The 4 words are referred to in the programs and in this report as words T, B, C, and D though occasionally word T may also be referred to as word G or as word A. As the 4 words for each new event come into the buffer, an address counter is counted up by 4 so that the next event will occupy the next 4 words in the buffer. This address counter works in a circular fashion so that it starts again at the beginning of the buffer when the end of it is reached. To prevent the data buffer from filling up and new events being written over old events which have not yet been processed the data break contains an up-down counter which counts up the new words coming in and can be counted down by the program as old words are taken out of the buffer and processed. If the data rate becomes fast enough that the data buffer fills up faster than the program can empty it out then each time the buffer is completely filled with unprocessed data the up-down counter overflows and this condition stops new data from coming in until some of the old data has been removed and the up-down counter counted down by the program.

The following machine instructions are used for program control of the data break:

BADCLR Break control ADdress counter CLeaR -- sets the address counter to the address of the first word in the buffer region so that the buffer will start at the position given by the hardware switches.

BABLE Break enABLE in Break control -- enables the data break and allows data to start coming in.

BDISAB Break DISABle in Break control -- disables the data break and stops the data from coming in.

UDCLR Up-Down CLeaR -- clears the up-down counter and issues an accepted pulse to the interface.

UDSOFL Up-Down counter Skip on OverFLow -- causes the computer to skip the next instruction if the up-down counter is in the overflowed condition.

UDSUB1 Up-Down counter, SUB 1 -- subtracts 1 from the up-down counter.

UDSUB2 Up-Down counter, SUB 2 -- subtracts 2 from the Up-Down counter.

UDSUB3 Up-Down counter, SUB 3 -- substracts 3 from the up-down counter.

All of the circuits except for the address counter are rest by the POWER CLEAR which accompanies the start switch on the computer and by the clear signal generated by the UDCLR pulse.

The four signals that the program is concerned with are denoted T, B, C, and D. The B, C, and D signals are the digitized amplitudes of the energy loss in a series of 3 detectors in a telescope arrangement. The incident particle first goes through B, then C, then D. It is possible for the signal to stop in B or C or D but not to penetrate them all. The electronics are set up so as to give a constant energy loss per channel in each detector. Thus the sum of energy loss in the telescope is B + C + D. If only the first two detectors are penetrated, D = 0 so that the total energy is just B + C. The T signal is derived from the time that it takes for the particle to go from the target to the B detector. This information is of interest only if the particle stops in B so that signals are not available from C and/or D.

PROGRAM DESCRIPTIONS

LITTLE HOOP

LITTLE HOOP is a minimum program for testing the data break. It flickers each event as a dot on the oscilloscope whose x and y coordinates are any two of the four words corresponding to the event. The user can choose which two of the words he wants plotted and which axis he wants each of them on. If data stops coming in for as long as a second or so the program rings a bell and starts over.

BIG HOOP

BIG HOOP is a modification of LITTLE HOOP which allows the user to change the words being plotted on either or both of the two coordinate axes while the program is still operating. This is accomplished by jumping to a subroutine COMMND from various strategic points in the program. COMMND is designed to be sensitive to certain special signals typed in from the keyboard and, upon receiving the proper signals, to change the words being plotted along the coordinate axes. It then returns control to BIG HOOP. If no signals are typed in or if the wrong signals are typed in, control is returned to BIG HOOP without changing the axes.

COMMND

COMMND is a subroutine which, if it is periodically called by BIG HOOP, allows the user to change certain prespecified words in core. It is also the prototype of more sophisticated routines which could be used to give the experimenter more capability for interaction with complicated data acquisition programming systems. Each time COMMND is called by the main program it checks to see if any new signals have been typed in from the keyboard.

If the user has not typed anything in, control is returned immediately to the main program. If something has been typed in COMMND checks to see if it was a "permitted" signal. If it was not, control is returned to the main program. If it was a permitted signal it then carries out the proper action. The present version of COMMND has two permitted signals, B and C. The signal C tells the program that the user wishes to change some location in core. Upon receiving a C, COMMND will print the C and jump to a subroutine called RDOCTL which then allows the user to type in the address of the location he wants to change as a four-digit octal number. When control returns from RDOCTL, COMMND then checks to see if the location that was typed in is one of the locations that it is permitted to change. The present version of COMMND is only allowed to change locations 0376 and 0377 which should contain the word numbers (0, 1, 2, 3 for words T, B, C, D respectively) to be plotted along the x and y axes respectively, but the vector of addresses of locations that it is permitted to change could easily be extended to allow the user to change other locations. After RDOCTL has returned control to COMMND, the latter waits for the user to type in either an O or a D. If the user types in an O then COMMND again jumps to RDOCTL so the user can type the new word he wants loaded as a four digit octal number. If the user types in a D then COMMND jumps to a subroutine called RDDEC which allows him to type in the new word as a four-digit decimal number. After the return from RDOCTL or RDDEC, whichever the case may be, COMMND prints out a carriage return and a line feed and them returns control to the main program. The other permitted signal, B, will cause COMMND to ring the bell on the teletype. This feature serves a double purpose. First, if the experimenter becomes bored during the experiment he can punch B and the ting of the bell will break the monotony. Second, it serves as an example of how COMMND can be extended to provide other commands which the experimenter can use to control the course of the experiment. (For example, if 6 other bells were attached to the teletype keys A, C, D, E, F, and G and if all seven bells were then tuned accordingly, then the experimenter could play simple melodies if he should get tired of the constant tone of B alone .)

RDOCTL

RDOCTL is a program which is called by COMMND to read four-digit octal numbers typed in by the user. It prints out each digit as it is typed, converts it to binary, and adds it to the binary word being accumulated, and after all four digits have been typed in it returns control to COMMND with the binary equivalent of the four-digit octal word in the AC.

RDDEC

RDDEC is a program which is called by COMMND to read four-digit decimal numbers typed in by the user. It prints out each digit as it is typed, converts it to binary, and adds it to the binary equivalent word being accumulated unless the digit typed in would cause an overflow in which case it gives a return to a section of COMMND which types a question mark and then returns control to the main program (BIG HOOP). Thus if the user attempted to type in the decimal number 4367 he would get instead 43? and a return to BIG HOOP. If the decimal number being typed in does not cause an overflow return, then control is returned to COMMND with the binary equivalent of the number in the AC.

RDIGIT and TYPEO

RDIGIT is a subroutine which is called by COMMND, RDOCTL, and RDDEC to read a character typed in from the keyboard. TYPEO is a subroutine called by COMMND, RDOCTL, and RDDEC to print out the characters that are typed in by the user.

OPERATING INSTRUCTIONS FOR LITTLE HOOP,

A MINIMUM VERSION OF PETER T. HOOPER

LITTLE HOOP uses a 1000 word data buffer beginning at loc. 1000,

WORD-X and WORD-Y are selected by switches (0, 1, 2, 3).

WORD-X is plotted on X axis and displayed in the AC.

WORD-Y is plotted on Y axis and displayed in the MQ.

If there are no interrupts in about 1.0 seconds, then the bell is tinged and the program restarted.

Data break control switches should be set as follows:

Origin of buffer,	1000.
Number of words,	4 (left SW, up, right SW, up).
Buffer length,	256.

When all the switches are set properly, proceed as follows:

(1) Load HOOP from RIM tape.

(2) Start in loc. 3000.

(3) Prog. halts for WORD-X (0, 1, 2, or 3) to be keyed in. Key it in and hit continue.

(4) Prog. halts for WORD-Y. Key it in and hit continue.

OPERATING INSTRUCTIONS

for

BIG HOOP

BIG HOOP is a command program version of Peter T. Hooper. Start program at 200. Bell will ting and program starts. Set all data break control switches in standard positions. Original option is to flicker C on X-axis vs D on Y-axis. Y shows in AC, X shows in MQ.

In order to modify which words flicker, use the typewriter. Type

```
C 0376 ∅ 0000 to make X flicker word 0  (T)
C 0376 ∅ 0001 to make X flicker word 1  (B)
C 0376 ∅ 0002 to make X flicker word 2  (C)
C 0376 ∅ 0003 to make X flicker word 3  (D).
```

In case of illegal command, computer will type "?" and you can try again. To change the word which makes Y flicker, do the same except use 0377 instead of 0376.

ORNL DWG. 66-3810

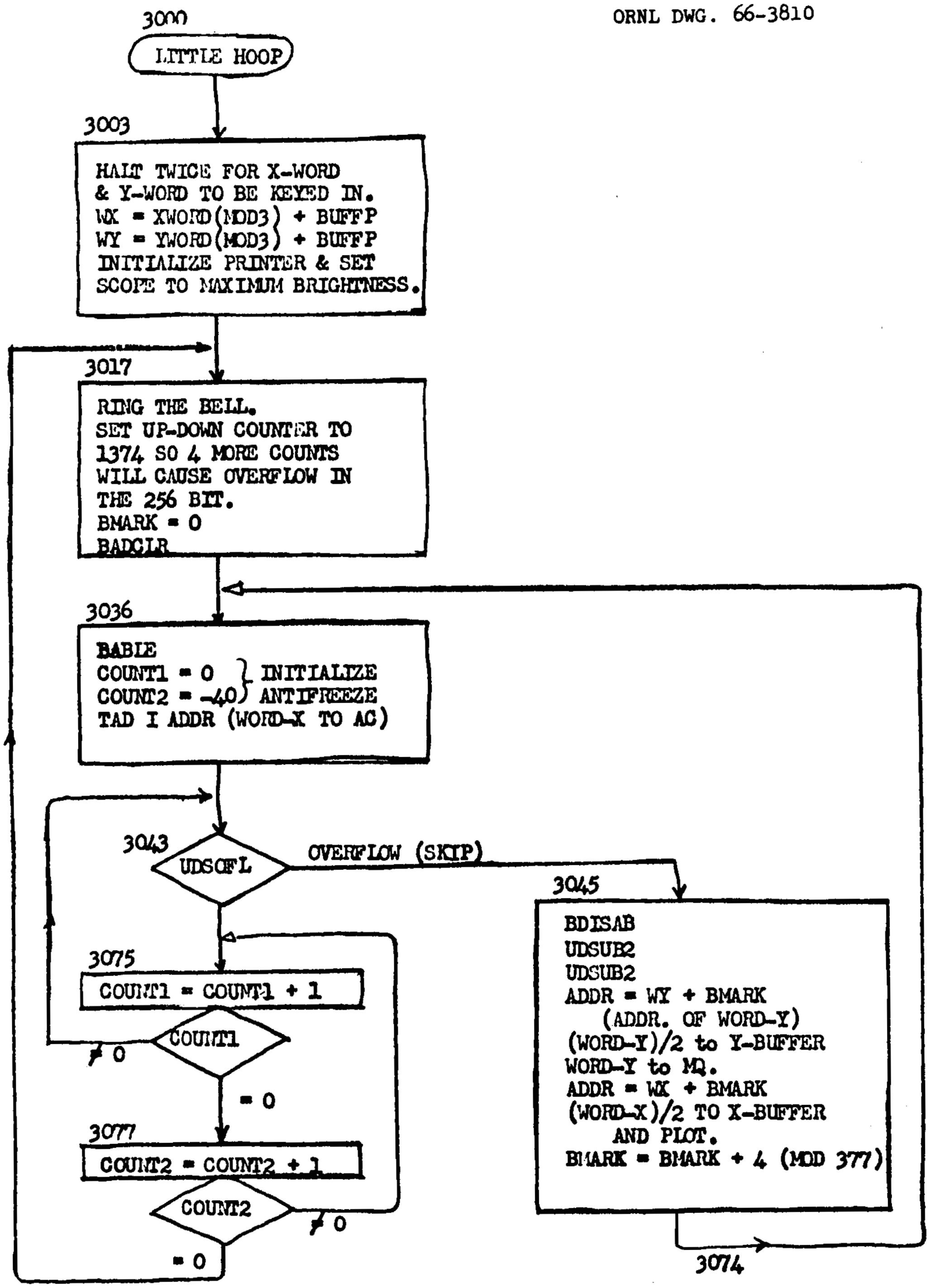

Fig. 1. Flow Chart of LITTLE HØØP.

ORNL DWG. 66-3811

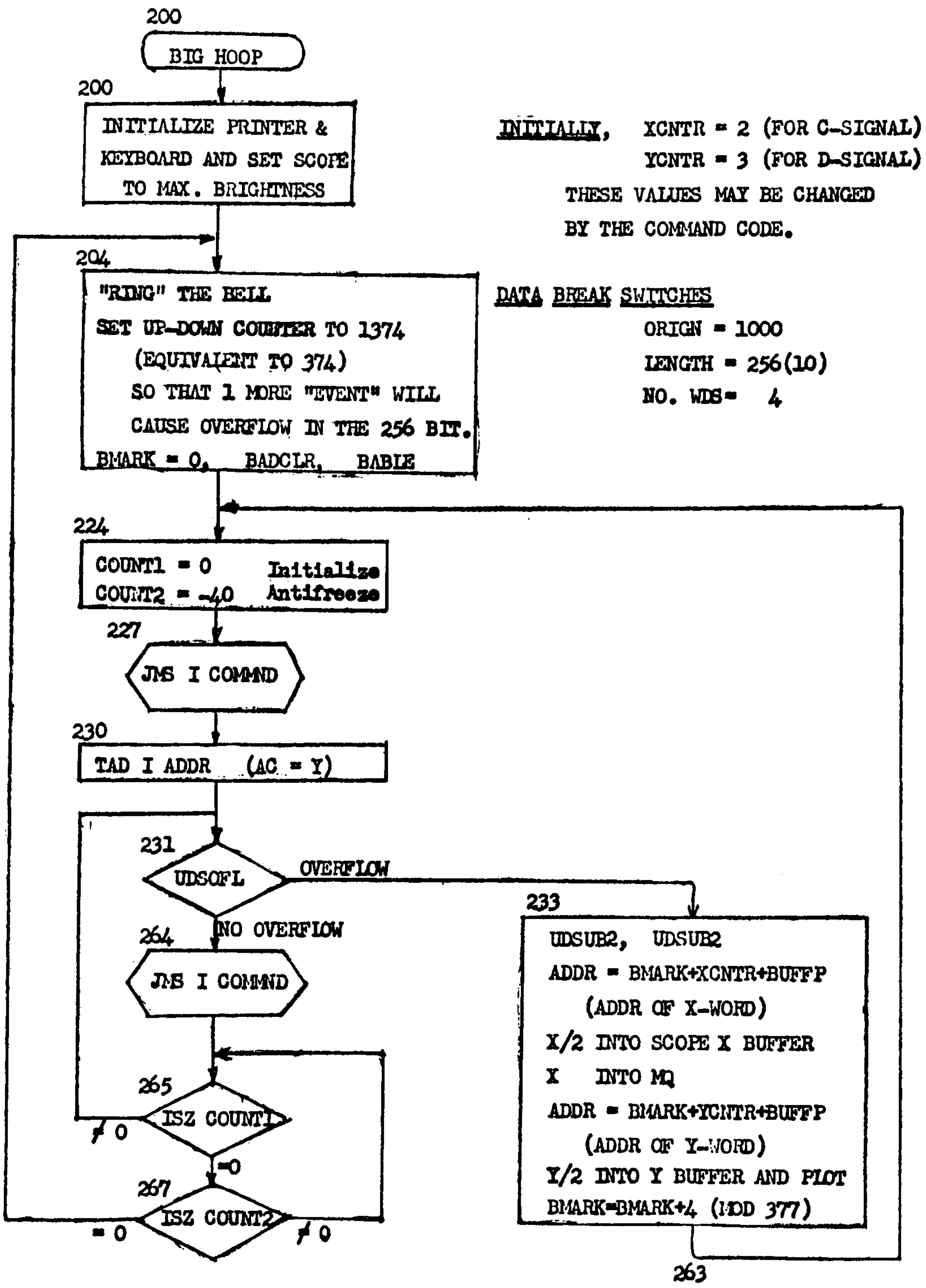

Fig. 2. Flow Chart of BIG HØØP.

ORNL DWG. 66-3812

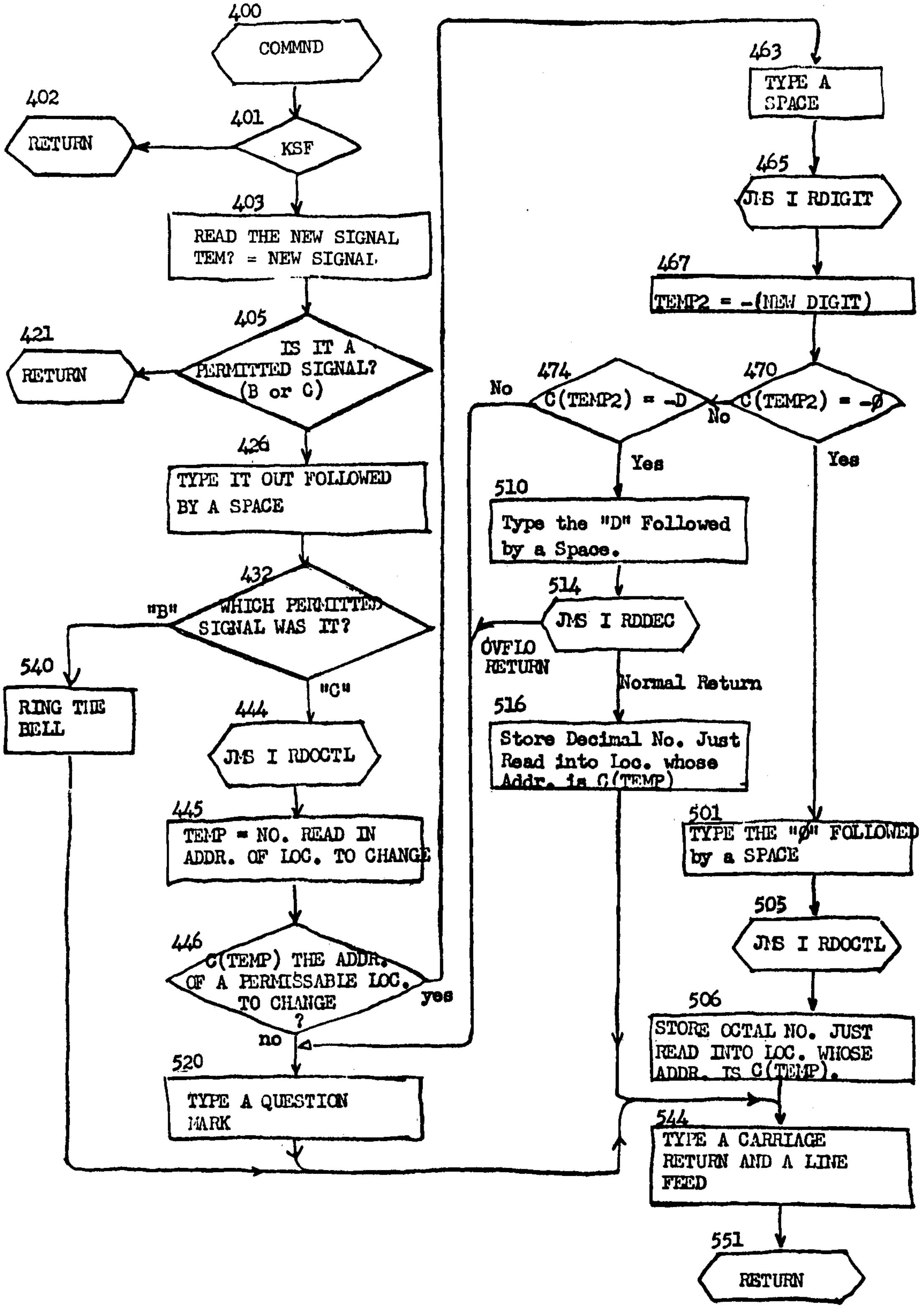
400
COMMND
402
RETURN
401
KSF
403
READ THE NEW SIGNAL
TEMP = NEW SIGNAL
405
IS IT A PERMITTED SIGNAL? (B or C)
421
RETURN
426
TYPE IT OUT FOLLOWED BY A SPACE
432
WHICH PERMITTED SIGNAL WAS IT?
"B"
"C"
540
RING THE BELL
444
JMS I RDOCTL
445
TEMP = NO. READ IN ADDR. OF LOC. TO CHANGE
446
C(TEMP) THE ADDR. OF A PERMISSABLE LOC. TO CHANGE ?
yes
no
520
TYPE A QUESTION MARK
463
TYPE A SPACE
465
JMS I RDIGIT
467
TEMP2 = -(NEW DIGIT)
470
C(TEMP2) = -Ø
No
Yes
474
C(TEMP2) = -D
No
Yes
510
Type the "D" Followed by a Space.
514
JMS I RDDEC
OVFLO RETURN
Normal Return
516
Store Decimal No. Just Read into Loc. whose Addr. is C(TEMP)
501
TYPE THE "Ø" FOLLOWED by a SPACE
505
JMS I RDOCTL
506
STORE OCTAL NO. JUST READ INTO LOC. WHOSE ADDR. IS C(TEMP).
544
TYPE A CARRIAGE RETURN AND A LINE FEED
551
RETURN

Fig. 3. Flow Chart of COMMND.

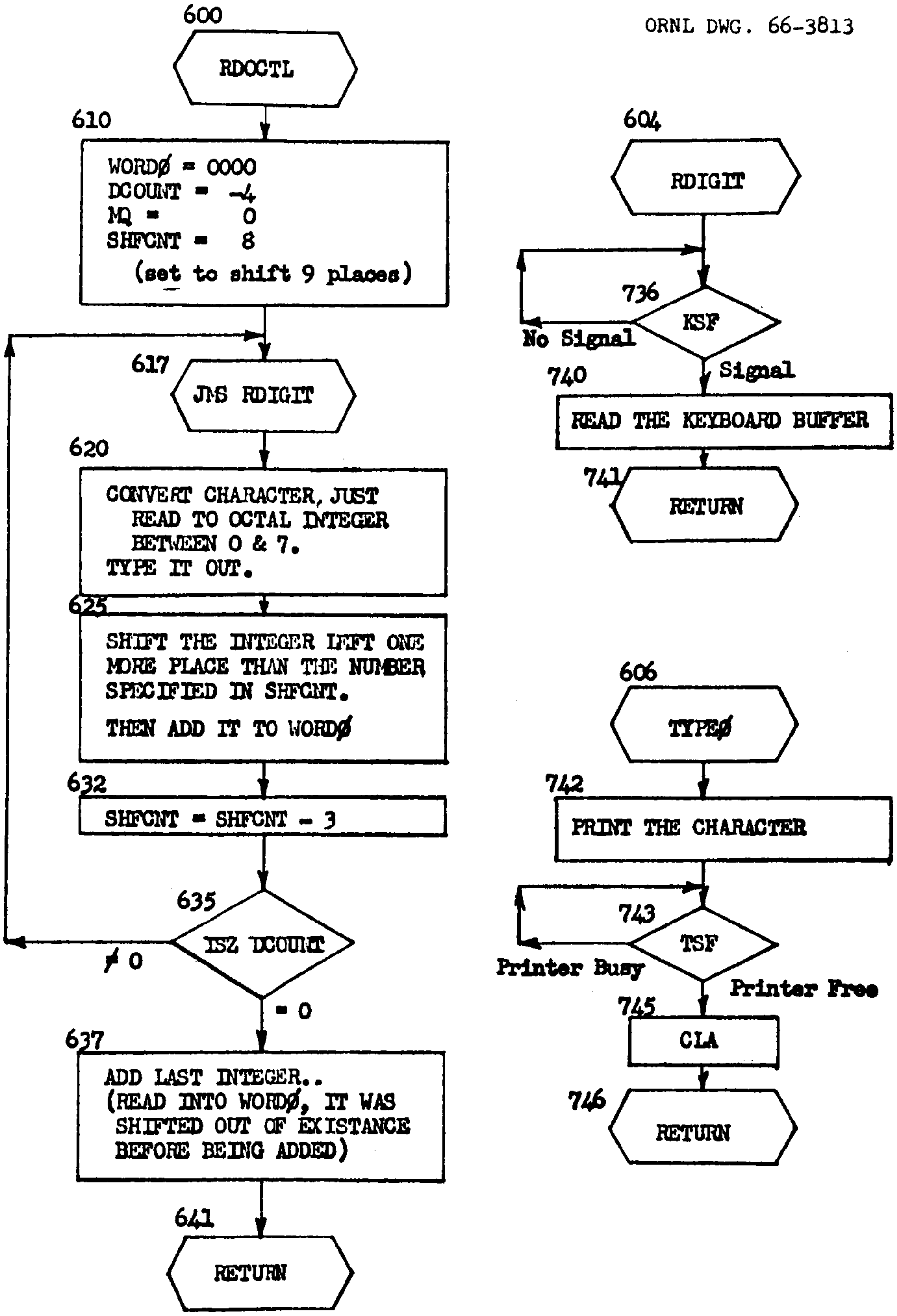

Fig. 4. Flow Charts of RDØLTL, RDIGIT, TYPRØ.

ORNL DWG. 66-3814

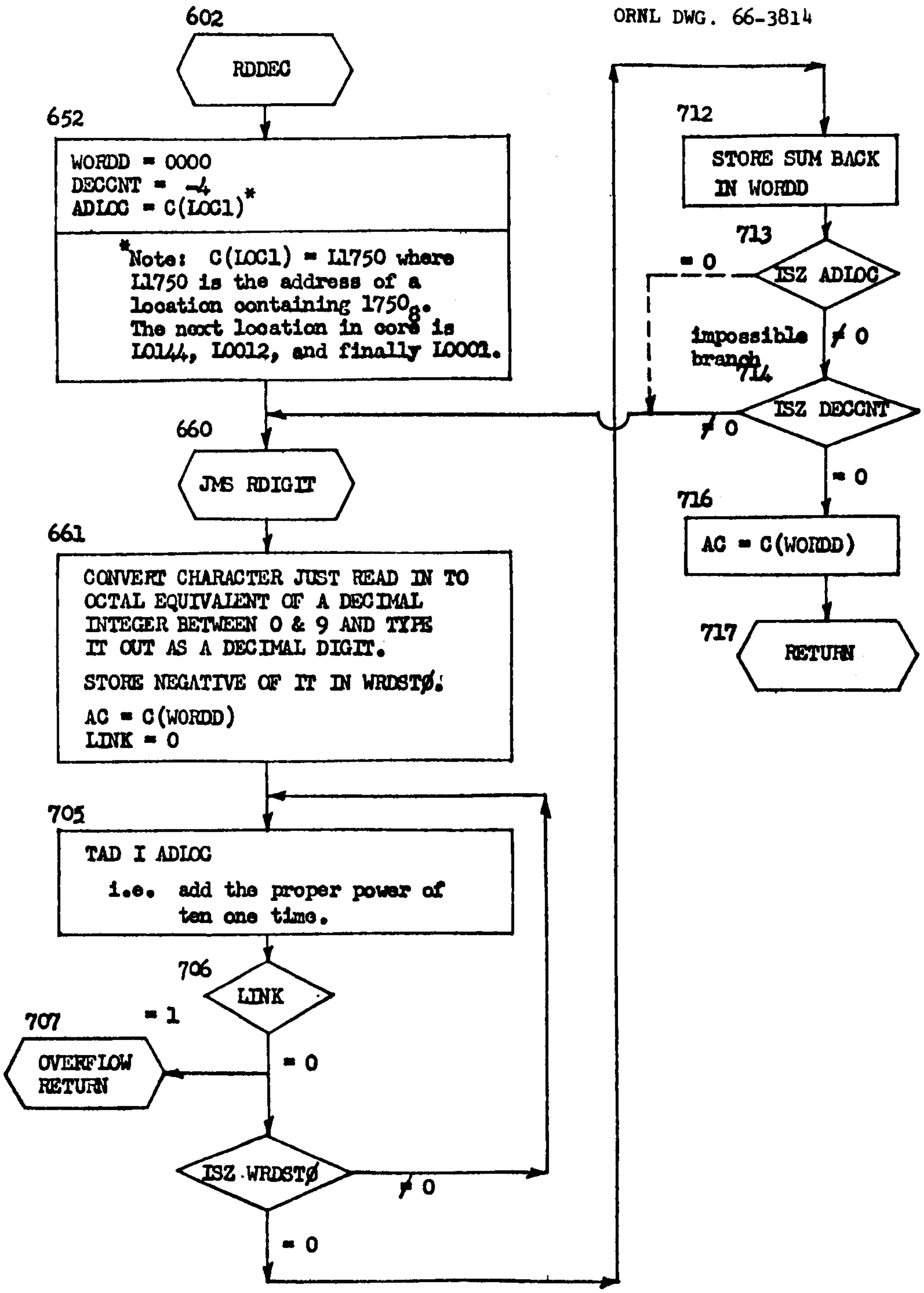
602
RDDEC
652
WORDD = 0000
DECCNT = -4
ADLOC = C(LOC1)*
*Note: C(LOC1) = L1750 where L1750 is the address of a location containing 1750_8. The next location in core is L0144, L0012, and finally L0001.
660
JMS RDIGIT
661
CONVERT CHARACTER JUST READ IN TO OCTAL EQUIVALENT OF A DECIMAL INTEGER BETWEEN 0 & 9 AND TYPE IT OUT AS A DECIMAL DIGIT.
STORE NEGATIVE OF IT IN WRDSTØ.
AC = C(WORDD)
LINK = 0
705
TAD I ADLOC
i.e. add the proper power of ten one time.
706
LINK
= 1
707
OVERFLOW RETURN
= 0
ISZ WRDSTØ
≠ 0
= 0
712
STORE SUM BACK IN WORDD
713
ISZ ADLOC
= 0
impossible branch
≠ 0
714
ISZ DECCNT
≠ 0
= 0
716
AC = C(WORDD)
717
RETURN

Fig. 5. Flow Chart of RDDEC.

```
                 / LITTLE HOOP, A MINIMUM VERSION OF PETER T. HOOPER
                 /        USES A 256 WORD DATA BUFFER BEGINNING AT LOC. 1000.
                 /        OCT, 22, 1966    R. BURRUS
                 /
                 /WORD-X AND WORD-Y ARE SELECTED BY SWITCHES (0,1,2,3)
                 /
                 /WORD-X IS PLOTED ON X AXIS AND DISPLAYED IN THE AC.
                 /
                 /WORD-Y IS PLOTED ON Y AXIS AND DISPLAYED IN THE MQ.
                 /
                 /IF THERE ARE NO INTERUPTS IN ABOUT 1.0 SECONDS, THEN
                 /      THE BELL IS TINGED AND THE PROGRAM RESTARTED.
                 /
                 /DATA BREAK CONTROL SWITCHES SHOULD BE SET AS FOLLOWS
                 /        ORIGN OF BUFFER, 1000
                 /        NUMBER OF WORDS,     4  (LEFT SW, UP, RIGHT SW, UP)
                 /        BUFFER LENGTH,     256
                 /
                 /
                 *3000
3000  7000                OPR
3001  7000                OPR
3002  7000                OPR
3003  7402       PETE,    HLT                 /GET WORD=X AND WORD-Y
3004  7604                LAS
3005   316                AND       L0003
3006  1310                TAD       BUFFP
3007  3313                DCA       WX
3010  7402                HLT
3011  7604                LAS
3012   316                AND       L0003
3013  1310                TAD       BUFFP
3014  3314                DCA       WY
3015  6046                TLS                 /INITIALZE PRINTER AND
3016  6077                DSB+3               /SET BRIGHTNESS TO MAX
3017  7200       TING,    CLA
3020  1311                TAD       BELL
3021  6041                TSF
3022  5221                JMP       .-1
3023  6046                TLS
3024  6324                UDCLR               /CLEAR THE UP-DOWN COUNTER AND SUBTRACT
3025  7200                CLA                 /FROM IT 404 TIMES UNTIL IT READS 1374.
3026  1312                TAD       M126      /THEN FOUR MORE COUNTS WILL CAUSE AN
3027  3303                DCA       COUNT1    /OVERFLO IN THE  256 BIT.
3030  6337                UDSUB3
3031  2303                ISZ       COUNT1
3032  5230                JMP       .-2
3033  6333                UDSUB2

3034  3305                DCA       BMARK     /ZERO BUFFER MARKER AND DATA ADDRESS COUNTER
3035  6311                BADCLR
3036  6312       LOOP,    BABLE
3037  3303                DCA       COUNT1    /SET UP THE ANTI-FRZ INITIALIZATION
3040  1307                TAD       M40
3041  3304                DCA       COUNT2
3042  1702                TAD  I    ADDR      /AC = WORD-X (NOT DEFINED 1ST PASS)
3043  6321       WAIT,    UDSOFL
3044  5275                JMP       DELAY
3045  6314                BDISAB
3046  6333                UDSUB2
```

```
3047  6333            JDSUB2
3050  7200            CLA
3051  1305            TAD       BMARK
3052  1314            TAD       WY
3053  3302            DCA       ADDR
3054  1702            TAD   I   ADDR
3055  7010            RAR
3056  6063            DYL                  /LOAD WORD-Y/2 FOR SCOPE (1023 MAX)
3057  7004            RAL
3060  7421            MQL                  /MQ = WORD-Y
3061  1305            TAD       BMARK
3062  1313            TAD       WX
3063  3302            DCA       ADDR
3064  1702            TAD   I   ADDR
3065  7010            RAR
3066  6057            DXS                  /LOAD WORD-X/2 FOR SCOPE (1023 MAX)
3067  7200            CLA
3070  1305            TAD       BMARK
3071  1315            TAD       P4
3072   306            AND       L0377
3073  3305            DCA       BMARK
3074  5236            JMP       LOOP
3075  2303  DELAY,    ISZ       COUNT1
3076  5243            JMP       WAIT
3077  2304            ISZ       COUNT2
3100  5275            JMP       .-3
3101  5217            JMP       TING
3102     0  ADDR,     0000
3103     0  COUNT1,      0
3104     0  COUNT2,      0
3105     0  BMARK,       0
3106   377  L0377,     377
3107  7740  M40,       -40
3110  1000  BUFFP,    1000
3111   207  BELL,      207
3112  7652  M126,     -126
3113     0  WX,          0
3114     0  WY,          0
3115     4  P4,          4
3116     3  L0003,       3

SYMBOL TABLE
ADDR      3102
BELL      3111
BMARK     3105
BUFFP     3110
COUNT1    3103
COUNT2    3104
DELAY     3075
L0003     3116
L0377     3106
LOOP      3036
M126      3112
M40       3107
P4        3115
PFTE      3003
TING      3017
WAIT      3043
WX        3113
WY        3114
```

```
DUPLICATE TAGS
NØNE
UNDEFINED SYMBØLS
NØNE
```

```
                    / PERMANENT SYMBOLS FOR P.T. HOOPER AND COMMAND CODE.
                    /
                    *100
                    /
        100    400  COMMND, 0400     /LOC. = 100
        101    600  RDOCTL, 0600     /LOC. = 101
        102    602  RDDEC,  0602     /LOC. = 102
        103    604  RDIGIT, 0604     /LOC. = 103
        104    606  TYPEO,  0606     /LOC. = 104
                    /

        SYMBOL TABLE
        COMMND     100
        RDDEC      102
        RDIGIT     103
        RDOCTL     101
        TYPEO      104

        DUPLICATE TAGS
        NONE

        UNDEFINED SYMBOLS
        NONE
```

```
250  1377            TAD     YCNTR
251  1301            TAD     BUFFP
252  3272            DCA     ADDR
253  1672            TAD I   ADDR
254  7010            RAR
255  6067            DYS                           / LØAD Y/2 FØR SCØPE (1023 MAX)
256  7200            CLA
257  1275            TAD     BMARK                 / UPDATE BMARK
260  1277            TAD     L0004
261   276            AND     L0377
262  3275            DCA     BMARK
263  5224            JMP     LØØP
264  4500  DELAY,    JMS  I  CØMMND
265  2273            ISZ     CØUNT1
266  5231            JMP     WAIT
267  2274            ISZ     CØUNT2
270  5265            JMP     .-3
271  5204            JMP     TING                  / RESTART IF DELAY .GT. SECØND
272     0            ADDR,   0
273     0   CØUNT1,    0                           / FØR INITIALING UP-DØWN CØUNTER
274     0   CØUNT2,    0                           /     AND FØR DELAY TIMEING
275     0   BMARK,     0                           / KEEPS IN STEP WITH DATA
276   377   L0377,   377                           /     ADDRESS CØUNTER
277     4  L0004,   0004
300  7740   M40,     -40
301  1000   BUFFP,  1000
302   207   BELL,    207
303  7652            M126,   -126
           /
           /
           *376
376     2  XCNTR,   0002
377     3  YCNTR,   0003
           /

SYMBØL TABLE
ADDR        272
BELL        302
BMARK       275
BUFFP       301
CØMMND      100
CØUNT1      273
CØUNT2      274
DELAY       264
L0004       277
L0377       276
LØØP        224
M126        303
M40         300
PETE        200
TING        204
WAIT        231
XCNTR       376
YCNTR       377

DUPLICATE TAGS
NØNE

UNDEFINED SYMBØLS
NØNE
```

```
                / BIG HOOP, A COMMAND CODE VERSION OF PETER T. HOOPER
                /
                / A MINIMUM PROGRAM FOR TESTING THE DEC-MODEL 8 MULTIPLEXED
                / PARALEL DATA BREAK INTERFACE WITH A PULSE HEIGHT DIGITIZER
                /
                / IF THERE ARE NO INTERUPTS IN ABOUT 1.0 SECONDS, THEN
                /      THE BELL IS TINGED AND THE PROGRAM RESTARTED
                /
                / DATA BREAK CONTROL SWITCHES SHOULD BE SET AS FOLLOWS
                /      ORIGIN OF BUFFER, 1000
                /      NUMBER OF WORDS,    4   (LEFT SW. UP, RIGHT SW. UP)
                /      BUFFER LENGTH,    256
                /
                /      MAY 1966, W. R. BURRUS
                /         CHANGED OCT 1966,  B. W. RUST
                /
                COMMND=0100
                /
                *200
200   7200      PETE,   CLA                        / INITIALIZE PRINTER
201   6046              TLS                        /       AND KEYBOARD
202   6036              KRB
203   6077              DSB+3                      / SET BRIGHTNESS TO MAX
204   7200      TING,   CLA
205   1302              TAD     BELL               / TING
206   6041              TSF
207   5206              JMP     .-1
210   6046              TLS
211   6324              UDCLR                      / CLEAR THE UP-DOWN COUNTER
212   7200              CLA                        /       AND SUBTRACT UNTIL IT
213   1303              TAD     M126               /     READS 1374.  THEN FOUR
214   3273              DCA     COUNT1             /       MORE COUNTS WILL CAUSE
215   6337              UDSUB3                     /       AN OVERFLO IN THE
216   2273              ISZ     COUNT1             /       256 BIT.
217   5215              JMP     .-2
220   6333              UDSUB2
221   3275              DCA     BMARK              / SET MARKER AND DATA-ADDRESS
222   6311              BADCLR                     /       COUNTER TO ZERO
223   6312              BABLE
224   3273      LOOP,   DCA     COUNT1
225   1300              TAD     M40
226   3274              DCA     COUNT2
227   4500              JMS  I  COMMND
230   1672              TAD I   ADDR               / AC = Y (Y NOT DEFINED 1ST PASS)
231   6321      WAIT,   UDSOFL
232   5264              JMP     DELAY
233   6333              UDSUB2
234   6333              UDSUB2
235   7200              CLA
236   1275              TAD     BMARK
237   1376              TAD     XCNTR
240   1301              TAD     BUFFP
241   3272              DCA     ADDR
242   1672              TAD I   ADDR
243   7010              RAR
244   6053              DXL                        / LOAD X/2 FOR SCOPE (1023 MAX)
245   7004              RAL
246   7421              MQL                         / MQ = X
247   1275              TAD     BMARK
```

```
                / COMMAND CODE FOR USE WITH P.T. HOOPER
                /
                /POINTERS
                RDOCTL=0101
                RDDEC=0102
                RDIGIT=0103
                TYPEO=0104
                /
                *400
                /
400       0     COMMND, 0000
401    6031             KSF                 /SKIP IF SIGNAL HAS BEEN TYPED IN.
402    5600             JMP   I   COMMND    /   OTHERWISE RETURN.
403    6036             KRB                 /READ THE SIGNAL
404    3330             DCA       TEMP      /   AND STORE IT
405    1222             TAD       MLGL      /CHECK TO SEE IF SIGNAL IS A LEGAL ONE
406    3334             DCA       COUNT
407    1223             TAD       LGLO
410    3335             DCA       ADDR
411    7200     SLOOP,  CLA                 /LOOP TO CHECK WHETHER SIGNAL TYPED IN IS A
412    1735             TAD   I   ADDR      /  PERMITTED SYMBOL
413    1330             TAD       TEMP
414    7450             SNA
415    5226             JMP       ALLOWD    /THE SIGNAL WAS A PERMITTED ONE
416    2335             ISZ       ADDR
417    2334             ISZ       COUNT
420    5211             JMP       SLOOP
421    5600             JMP   I   COMMND     /RETURN IF THE SIGNAL WAS NOT A LEGAL ONE
                /
422    7776     MLGL.   7776                /NEG. OF NO. OF PERMITTED SYMBOLS
423     424     LGLO.   SYMB1               /1ST LOC, OF VECTOR CONTAINING (2)COMPLEMENTS
                /                           /    OF PERMITTED SYMBOLS.
424    7475     SYMB1,  7475                /NEG, OF ASCII CODE FOR C (0303)
425    7476             7476                /NEG, OF ASCII CODE FOR B (0302)
                /
                /
426    1330     ALLOWD, TAD       TEMP      /ECHO THE SIGNAL
427    4504             JMS   I   TYPEO
430    1327             TAD       SPACE     /   AND TYPE A SPACE
431    4504             JMS   I   TYPEO
432    1222             TAD       MLGL      /NOW TAKE THE APPROPRIATE ACTION
433    7041             CIA
434    1334             TAD       COUNT
435    1237             TAD       .+2
436    7410             SKP
437    5242             JMP       .+3
440    3241             DCA       .+1       /STORE COMPUTED GO TO
441       0             0000                /JMP .+1, JMP .+2, ... WILL GO HERE
442    5244             JMP       ALTER
443    5340             JMP       RING
                /
                /
                /   ROUTINE TO CHANGE A LOCATION IN CORE
                /
                /    THE LOCS, IT IS PERMISSABLE TO CHANGE ARE
                /                           LOC. 0376 CONTAINING DETEC. NO. FOR X-AXIS
                /                           LOC. 0377 CONTAINING DETEC. NO. FOR Y-AXIS
                /                                  THESE NOS. SHOULD LIE BETWEEN 0 AND 3.
                /
```

```
444  4501  ALTER.  JMS  I  RDOCTL  /READ 4 OCTAL DIGITS FROM KEYBOARD
445  3330          DCA     TEMP
446  1332          TAD     MCHNG   /NEG. OF NO. LOCS. PERMISSABLE TO CHANGE
447  3334          DCA     COUNT
450  1333          TAD     CHNGO   /1ST ADDRESS IN VECTOR CONTAINING ADDRESSES
451  3335          DCA     ADDR    /    OF LOCS. PERMISSABLE TO CHANGE.
452  1330  CKLOOP, TAD     TEMP
453  7041          CIA
454  1735          TAD  I  ADDR
455  7650          SNA CLA
456  5263          JMP     CHANGE
457  2335          ISZ     ADDR
460  2334          ISZ     COUNT
461  5252          JMP     CKLOOP
462  5320          JMP     WHAT    /IF CONTROL REACHES THIS POINT THE LOC. TYPED
           /                       /   IN WAS NOT ONE PERMISSABLE TO CHANGE. TYPE
           /                       /   A QUESTION MARK AND RETURN TO CALLING PROG.
463  1327  CHANGE, TAD     SPACE   /TYPE A SPACE AND READ ANOTHER CHARACTER FROM
464  4504          JMS  I  TYPEO   /   THE KEYBOARD. (SHOULD BE EITHER O OR D).
465  4503          JMS  I  RDIGIT
466  7041          CIA             /   STORE NEG. OF THE DIGIT
467  3331          DCA     TEMP2
470  1331          TAD     TEMP2
471  1325          TAD     OCODE   / ASCII CODE FOR O
472  7650          SNA CLA
473  5301          JMP     OCTAL
474  1331          TAD     TEMP2
475  1326          TAD     DCODE   / ASCII CODE FOR D
476  7650          SNA CLA
477  5310          JMP     DECMAL
500  5320          JMP     WHAT    /IF CONTROL REACHES HERE THE CHARACTER
           /                       /   WAS NEITHER O NOR D -- TYPE A QUESTION
           /                       /   MARK AND RETURN.
501  1325  OCTAL.  TAD     OCODE   /TYPE THE O.
502  4504          JMS  I  TYPEO   /
503  1327          TAD     SPACE   /       AND THEN A SPACE.
504  4504          JMS  I  TYPEO
505  4501          JMS  I  RDOCTL  /READ THE OCTAL CHANGE FROM THE KEYBOARD.
506  3730          DCA  I  TEMP    /STORE THE CHANGE
507  5344          JMP     RETURN  /  AND RETURN
           /
510  1326  DECMAL, TAD     DCODE   /TYPE THE D
511  4504          JMS  I  TYPEO
512  1327          TAD     SPACE
513  4504          JMS  I  TYPEO
514  4502          JMS  I  RDDEC   /READ THE DECIMAL CHANGE FROM THE KEYBOARD.
515  5320          JMP     WHAT    /  (OVERFLOW RETURN)
516  3730          DCA  I  TEMP    /  (NORMAL RETURN -- STORE THE CHANGE)
517  5344          JMP     RETURN
520  7200  WHAT,   CLA
521  1324          TAD     L0277   / 0277 IS ASCII CODE FOR QUESTION MARK.
522  4504          JMS  I  TYPEO
523  5344          JMP     RETURN
           /
           /
524   277  L0277.  0277    /  ASCII CODE FOR QUESTION MARK
525   317  OCODE.  0317    /  ASCII CODE FOR O.
526   304  DCODE,  0304    /  ASCII CODE FOR D.
527   240  SPACE.  0240    /  ASCII CODE FOR SPACE
530     0  TEMP,   0000
```

```
531      0  TEMP2,   0000
532   7776  MCHNG,   7776     / NEG. OF NO. LOCS. IT IS PERMISSABLE TO CHANGE.
533    536  CHNG0,   XCNTR    / FIST LOC. IN VECTOR OF ADDRESSES PERMISSABLE TO CHANGE
534      0  COUNT,   0000
535      0  ADDR,    0000
            /
            /VECTOR OF LOCATIONS WHICH CAN BE CHANGED
            /
536    376  XCNTR,   0376     /LOC. OF DETEC. NO. TO PLOT ON X-AXIS
537    377  YCNTR,   0377     /LOC. OF DETEC. NO. TO PLOT ON Y-AXIS
            /
            /
            /  ROUTINE TO RING THE BELL
            /
540   1343  RING,    TAD      BELL
541   4504           JMS  I   TYPE0
542   5344           JMP      RETURN
            /
543    207  BELL,    0207              /ASCII CODE FOR THE BELL
            /
            /
544   7200  RETURN,  CLA
545   1352           TAD      L0215    / 0215 IS ASCII CODE FOR CARRIAGE RETURN
546   4504           JMS  I   TYPE0
547   1353           TAD      L0212    / 0212 IS ASCII CODE FOR LINE FEED
550   4504           JMS  I   TYPE0
551   5600           JMP  I   COMMND
552    215  L0215,   0215     /  ASCII CODE FOR CARRIAG RETURN
553    212  L0212,   0212     /  ASCII CODE FOR LINE FEED.
            /
```

```
SYMBOL TABLE
ADDR      535
ALLOWD    426
ALTER     444
BELL      543
CHANGE    463
CHNG0     533
CKLOOP    452
COMMND    400
COUNT     534
DCODE     526
DECMAL    510
L0212     553
L0215     552
L0277     524
LGL0      423
MCHNG     532
MLGL      422
OCODE     525
OCTAL     501
RDDEC     102
RDIGIT    103
RDOCTL    101
RETURN    544
RING      540
SLOOP     411
SPACE     527
SYMB1     424
TEMP      530
```

```
TEMP2          531
TYPED          104
WHAT           520
XCNTR          536
YCNTR          537

DUPLICATE TAGS
NØNE

UNDEFINED SYMBØLS
NØNE
```

```
                  /
                  /COMMAND POP -- A SYSTEM OF SUBROUTINES FOR READING FROM KEYBOARD
                  /                  AND WRITING OUT ON THE TELETYPE.
                  /
                  /         ALL ENTRIES GIVEN FIRST WITH JUMPS TO CORRESPONDING SUBROUTINES
                  /
                  *600
                  /
600        0      RDOCTL, 0000
601     5210              JMP      RDOCT
602        0      RDDEC,  0000
603     5252              JMP      RDDCML
604        0      RDIGIT, 0000
605     5336              JMP      READ
606        0      TYPEO,  0000
607     5342              JMP      TYPE
                  /
                  /
                  /
                  /RDOCTL -- SUBROUTINE TO READ 4 CHARACTERS (OCTAL DIGITS) FROM THE
                  /           KEYBOARD AND COMPOSE THEM INTO 1 PDP-8 WORD WHICH IS
                  /           RETURNED IN THE AC.  THE PROGRAM ECHOES EACH CHARACTER
                  /           AS IT IS READ.
                  /
                  /        CALLING SEQUENCE          JMS   I  RDOCTL
                  /                                  XXXX           /CONTROL RETURNED HERE
                  /
610     7200      RDOCT.  CLA
611     3242              DCA      WORDO   /ZERO OUT WORDO
612     1243              TAD      M4
613     3244              DCA      DCOUNT
614     7421              MQL              /CLEAR THE MQ
615     1245              TAD      L0010   /STORE AN 8 IN SHIFT COUNTER
616     3227              DCA      SHFCNT  /   (THE 1ST CHARAC. SHIFTED 9 PLACES TO LEFT)
617     4204      RDLOOP, JMS      RDIGIT  /READ IN A CHARACTER
620      251              AND      L0007   /   AND CONVERT IT TO AN OCTAL DIGIT.
621     3246              DCA      TEMP1
622     1246              TAD      TEMP1
623     1247              TAD      L0260   /CONVERT DIGIT TO ASCII CODE
624     4206              JMS      TYPEO   /   AND TYPE IT OUT.
625     1246              TAD      TEMP1   /PICK UP THE DIGIT AND
626     7413              SHL              /   SHIFT IT LEFT PROPER NO. PLACES
627        0      SHFCNT, 0000
630     1242              TAD      WORDO   /   ADD IT INTO THE PDP-8 WORD BEING
631     3242              DCA      WORDO   /   COMPOSED.
632     1227              TAD      SHFCNT  /SET SHIFT COUNTER
633     1250              TAD      M3      /    FOR NEXT DIGIT.
634     3227              DCA      SHFCNT
635     2244              ISZ      DCOUNT
636     5217              JMP      RDLOOP
637     1242              TAD      WORDO
640     1246              TAD      TEMP1
641     5600              JMP   I  RDOCTL
                  /
642        0      WORDO,  0000
643     7774      M4,     7774
644        0      DCOUNT, 0000
645       10      L0010,  0010
646        0      TEMP1,  0000
```

```
647   260  L0260,  0260
650  7775  M3,     7775
651     7  L0007,  0007
           /
           /
           /
           /RDDEC -- SUBROUTINE TO READ 4 CHARACTERS (DECIMAL DIGITS) FROM THE
           /         KEYBOARD AND COMPOSE THEM INTO 1 PDP-8 WORD WHICH IS
           /         RETURNED IN THE AC.  THE PROGRAM ECHOES EACH CHARACTER
           /         AS IT IS READ.  THE PROGRAM RETURNS IN AN OVERFLOW
           /         RETURN IF THE NUMBER TYPED IN EXCEEDS 4095 (10) = 7777 (8).
           /
           /   CALLING SEQUENCE     JMS I RDDEC
           /                        XXXX          /OVERFLOW RETURN
           /                        XXXX          /NORMAL RETURN
           /
652  7200  RDDCML, CLA
653  3334          DCA      WORDD   /ZERO OUT WORDD (WILL ACCUM. THE PDP-8 WORD IN
654  1331          TAD      M0004   /                          WORDD)
655  3332          DCA      DECCNT
656  1322          TAD      LOC1
657  3333          DCA      ADLOC
660  4204  RDLP,   JMS      RDIGIT  /READ IN A CHARACTER (DECIMAL DIGIT).
661   327          AND      L0017   /   ANYTHING WHICH IS NOT ACTUALLY A
662  3335          DCA      WRDSTO  /   DECIMAL DIGIT IS CONVERTED TO SOME
663  1335          TAD      WRDSTO  /   DECIMAL DIGIT.  FIRST MASK OFF ALL
664  7041          CIA              /   BUT LAST 4 BINARY BITS.  IF THIS
665  1321          TAD      L0011   /   MASKED VALUE EXCEEDS 9 SET IT
666  7700          SMA CLA          /   EQUAL TO 9.
667  5272          JMP      CONT
670  1321          TAD      L0011
671  3335          DCA      WRDSTO
672  7200  CONT,   CLA
673  1335          TAD      WRDSTO  /TYPE OUT THE DECIMAL DIGIT.
674  1330          TAD      L260
675  4206          JMS      TYPEO
676  1335          TAD      WRDSTO
677  7450          SNA
700  5313          JMP      CONT2
701  7041          CIA
702  3335          DCA      WRDSTO
703  1334          TAD      WORDD   /CONVERT THE DIGIT TO OCTAL EQUIVALENT
704  7100          CLL
705  1733  CONLP,  TAD   I  ADLOC   /  AND ADD IT INTO WORDD.
706  7430          SZL
707  5602          JMP   I  RDDEC   /IN CASE OF OVERFLOW (OVERFLOW RETURN)
710  2335          ISZ      WRDSTO
711  5305          JMP      CONLP
712  3334          DCA      WORDD
713  2333  CONT2,  ISZ      ADLOC
714  2332          ISZ      DECCNT
715  5260          JMP      RDLP
716  1334          TAD      WORDD   /RETURN WITH THE PDP-8 WORD IN AC
717  2202          ISZ      RDDEC   /    (NORMAL RETURN)
720  5602          JMP   I  RDDEC
           /
721    11  L0011,  0011             /  =9(10)
722   723  LOC1,   L1750
723  1750  L1750,  1750             / =1000(10)
724   144  L0144,  0144             / = 100(10)
```

```
725     12  L0012,  0012             / =  10(10)
726      1  L0001,  0001
727     17  L0017,  0017
730    260  L260,   0260
731   7774  M0004,  7774
732      0  DECCNT, 0000
733      0  ADLOC,  0000
734      0  WORDD,  0000
735      0  WRDSTO, 0000
            /
            /
            /
            /RDIGIT -- A SUBROUTINE TO READ A CHARACTER FROM THE KEYBOARD.
            /           THE CHARACTER IS RETURNED IN THE AC.  CONTROL IS
            /           RETURNED TO NEXT SEQUENTIAL LOCATION IN THE CALLING
            /           PROGRAM
            /
736   6031  READ,   KSF
737   5336          JMP     .-1
740   6036          KRB
741   5604          JMP  I  RDIGIT
            /
            /
            /
            /TYPEO -- TYPES OUT THE CONTENTS OF THE AC WHEN CALLED.
            /     THE AC IS CLEARED BEFORE RETURNING TO NEXT SEQUENTIAL
            /     LOCATION IN THE CALLING PROGRAM.
            /
742   6046  TYPE,   TLS
743   6041          TSF
744   5343          JMP     .-1
745   7200          CLA
746   5606          JMP  I  TYPEO
            /
            /

SYMBOL TABLE
ADLOC       733
CONLP       705
CONT        672
CONT2       713
DCOUNT      644
DECCNT      732
L0001       726
L0007       651
L0010       645
L0011       721
L0012       725
L0017       727
L0144       724
L0260       647
L1750       723
L260        730
LOC1        722
M0004       731
M3          650
M4          643
RDDCML      652
RDDEC       602
RDIGIT      604
```

```
RDLØØP      617
RDLP        660
RDØCT       610
RDØCTL      600
READ        736
SHFCNT      627
TEMP1       646
TYPE        742
TYPEØ       606
WØRDD       734
WØRDØ       642
WRDSTØ      735

DUPLICATE TAGS
NØNE

UNDEFINED SYMBØLS
NØNE
```

```
     5 MINUTES, 13 SECONDS,          980  LINES.
END JOB 907.
```

Internal Distribution

1-3.	L. S. Abbott
4.	C. L. Allen (CTC)
5.	F. E. Bertrand
6.	A. A. Brooks (CTC)
7-9.	W. R. Burrus
10.	R. E. Funderlic (CTC)
11.	T. A. Love
12.	F. Madden
13.	F. C. Maienschein
14.	E. McDaniel
15-19.	R. W. Peelle
20.	F. G. Perey
21.	B. W. Rust
22.	L. W. Weston
23.	G. Dessauer (consultant)
24.	B. C. Diven (consultant)
25.	M. H. Kalos (consultant)
26.	L. V. Spencer (consultant)
27-28.	Central Research Library
29.	Document Reference Section
30-230.	Laboratory Records Department
231.	Laboratory Records ORNL RC
232.	ORNL Patent Office

External Distribution

233-247.	Division of Technical Information Extension (DTIE)
248.	Division of Research and Development (ORO)